IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
MIDLAND/ODESSA DIVISION

VIRTAMOVE, CORP., Plaintiff, v. ORACLE CORP., Defendant.	Case No. 7:24-CV-00339-ADA **JURY TRIAL DEMANDED**

DEFENDANT ORACLE CORP.'S MOTION TO DISMISS PLAINTIFF'S AMENDED COMPLAINT PURSUANT TO FED. R. CIV. P. 12(b)(6)

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Statutes

Other Authorities

I. INTRODUCTION

In its motion to dismiss VirtaMove's original complaint, Oracle demonstrated that the Asserted Patents are unpatentable because they claim an abstract idea in violation of 35 U.S.C. § 101. Dkt. 13. Instead of opposing that motion, VirtaMove amended its complaint to add a handful of paragraphs with attorney assertions that "[t]he technology… was not routine or conventional" and "[t]he technology claimed is not directed to an abstract idea." *See* Dkt. 21 ("FAC") ¶¶ 12, 14, 23 and 25. As a matter of law, such "[c]onclusory allegations, or those 'wholly divorced' from the claims or the specification, cannot defeat a motion to dismiss" for patent ineligibility. *Beteiro, LLC v. DraftKings Inc.*, 104 F.4th 1350, 1358 (Fed. Cir. 2024).

VirtaMove's amended complaint does not change the fact—established in Oracle's original motion—that the Asserted Patents are merely directed to desired results in functional terms, not how those results are achieved. Specifically, the '814 Patent is directed to the abstract idea of replicating software components into "containers" so they can be used in different environments, and the '058 Patent is directed to the equally abstract idea of copying shared resources so they can be used by different entities for different purposes at the same time. Both patents recite generic, well-known, prior art components (such as processors and operating systems) used in a conventional manner. The alleged benefits of the patents that VirtaMove recites cannot confer eligibility because they merely flow from performing the abstract ideas in a computing context, not from improvements to computers themselves.

Courts routinely invalidate such claims as patent-ineligible abstract ideas. Because VirtaMove reviewed Oracle's original motion and amended its complaint with averments that cannot salvage the Asserted Patents, further amendments would be futile. VirtaMove's Amended Complaint should be dismissed with prejudice.

II. LEGAL STANDARDS

The Federal Circuit has consistently held that "[s]ubject matter eligibility under § 101 may be determined at the Rule 12(b)(6) stage of a case." *ChargePoint, Inc. v. SemaConnect, Inc.*, 920 F.3d 759, 765 (Fed. Cir. 2019). To determine eligibility at the pleading stage, a court should have a "full understanding of the basic character of the claimed subject matter," but claim construction is not required. *Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat'l Ass'n*, 776 F.3d 1343, 1349 (Fed. Cir. 2014).

"[P]atent eligibility can be determined at the Rule 12(b)(6) stage" "when there are no factual allegations that, taken as true, prevent resolving the eligibility question as a matter of law." *Aatrix Software, Inc. v. Green Shades Software, Inc.*, 882 F.3d 1121, 1125 (Fed. Cir. 2018). A patentee cannot avoid the issue simply by pleading that the claims are not directed to an abstract idea or include patent-eligible subject matter. Factual allegations must be tethered to the patent's claims—conclusory or boilerplate allegations need not be accepted as true. *See IPA Techs., Inc. v. Amazon.com, Inc.*, 352 F. Supp. 3d 335, 349 (D. Del. 2019) (not treating as true "boilerplate allegations that the claims are directed to new computer functionality and improvements to technological processes… where those allegations contradict the language of the claims and specification"); *see also Beteiro*, 104 F.4th at 1358.

III. ARGUMENT

As explained in Oracle's original motion, the Asserted Patents are directed to patent-ineligible subject matter under 35 U.S.C. § 101. VirtaMove's amended complaint does not and cannot cure this fundamental failure. Thus, the Court should dismiss the amended complaint with prejudice.

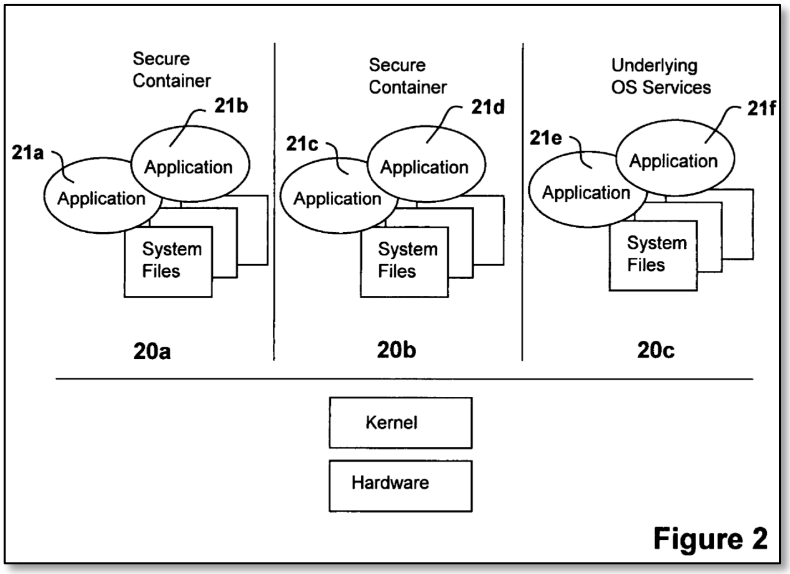
A. The ’814 Patent Claims Are Invalid For Claiming Ineligible Subject Matter

The ’814 Patent is invalid under Section 101 because it fails *Alice*’s two-step test. It fails ***Step One*** because the claims are directed to the abstract idea of replicating software components into “containers” so they can be used in different environments. Courts have routinely found similar claims unpatentable. It fails ***Step Two*** because it recites conventional computer components and does not add an inventive concept.

1. Background

The ’814 Patent is entitled “System for Containerization of Application Sets,” but the sole claim specifically asserted in the amended complaint—claim 1—is a method claim with a single “storing” step. ’814 Patent, 17: 41-61. Although claim 1 is wordy, the alleged invention is quite simple. The ’814 Patent describes a way of organizing data on a computer system into what the patent calls “containers.” Notably, the patent does not purport to have invented containers. *See, e.g.*, *id.*, 1:51-2:12 (“Background of the Invention” describing “existing solutions” including containers and identifying prior art product that provides “separation of an application from the underlying operating system.”).

As the patent explains, a container is simply “[a]n aggregate of files required to successfully execute a set of software applications on a computing platform.” ’814 Patent, 2:23-25. In other words, while the computer system has shared hardware (processors, memory, *etc.*), “underlying OS [operating system] services” (software managing the system’s shared resources), and a kernel (a privileged part of the operating system capable of directly controlling the hardware), each container has the “application” and “[operating] system files” necessary for that application to execute. That way, the application in the container does not depend on files in the computing system’s operating system because each container contains “all files required to successfully execute a set of software applications on a computing platform.” *Id.*, 7:26-27. Because these system files are now copied and stored in the containers, the files are organized in the container's "root file system." *Id.*, 11:35-39. Figure 2 illustrates a system with "containers" of applications and the associated files installed on a server with the underlying kernel and hardware:



The '814 Patent admits that before it was filed, "[c]omputer systems [were] designed in such a way that application programs share common resources." '814 Patent, 1:20-21. Thus, the patent concedes "[t]here are existing solutions that address the single use nature of computer systems." *Id.*, 1:51-52. The patent asserts that the "key difference" between the prior art and the purported invention "is that in the former an operating system, including files and a kernel, must be deployed for each application while the latter only requires one operating system regardless of the number of application containers deployed." *Id.* 1:56-61. This is nothing more than the abstract concept of replicating and organizing items in an obvious and convenient manner.

2. The '814 Patent Is Directed To An Abstract Idea

Representative claim 1 of the '814 Patent claims the abstract idea of replicating and organizing data into containers so applications can be used in different operating environments.

> 1. In a system having a plurality of servers with operating systems that differ, operating in disparate computing environments, wherein each server includes a

> processor and an operating system including a kernel a set of associated local system files compatible with the processor, a method of providing at least some of the servers in the system with secure, executable, applications related to a service, wherein the applications are executed in a secure environment, wherein the applications each include an object executable by at least some of the different operating systems for performing a task related to the service, the method comprising:
>
> storing in memory accessible to at least some of the servers a plurality of secure containers of application software, each container comprising one or more of the executable applications and a set of associated system files required to execute the one or more applications, for use with a local kernel residing permanently on one of the servers;
>
> wherein the set of associated system files are compatible with a local kernel of at least some of the plurality of different operating systems, the containers of application software excluding a kernel,
>
> wherein some or all of the associated system files within a container stored in memory are utilized in place of the associated local system files that remain resident on the server,
>
> wherein said associated system files utilized in place of the associated local system files are copies or modified copies of the associated local system files that remain resident on the server, and
>
> wherein the application software cannot be shared between the plurality of secure containers of application software, and
>
> wherein each of the containers has a unique root file system that is different from an operating system's root file system.

'814 Patent, 17:30-61. To the extent the preamble is limiting, it merely recites generic computing components such as a "server" with "a processor and an operating system" capable of executing an application in a container. There is only one step in this method claim (highlighted yellow), which is storing containers that include an application and the files required to execute that application. As a whole, representative claim 1 is written in highly generic terms, focusing on "an abstract end-result," not "'a specific means or method' for improving technology." *RecogniCorp, LLC v. Nintendo Co.*, Ltd., 855 F.3d 1322, 1326 (Fed. Cir. 2017) (citation omitted).

The abstract nature of representative claim 1 can be illustrated by way of an analogy to a shopping mall food court that offers different cuisines (*e.g.*, pizza, sushi, *etc.*). The food court might have a single, large kitchen/pantry that contains all of the equipment and ingredients used to make all the styles of food on the menu. That kitchen could employ different chefs (a pizza cook, sushi chef, *etc.*) who each prepare their style of food, using a single shared stove, shared cookware, and shared ingredients. This is similar to the admitted prior art where the computer and its operating system (the large kitchen) must contain all of the resources (the ingredients and equipment) for each application (chef) to execute (make their style of food).

This approach has obvious problems. For example, chefs would compete for shared resources or there might be cross-contamination problems. An obvious solution is to have separate food stalls, each specializing in its own cuisine, where each chef would each have what they need (and only what they need) to separately prepare their own food. While each food stall might have some of the same condiments (salt and pepper shakers), each stall could also store and serve its own modified versions (parmesan cheese at one stall, soy sauce at another).

In the parlance of the ’814 Patent, a “container” is the food stall and an “application” is the chef. ’814 Patent, 17:42. While there is a pantry area with commonly used ingredients (“associated local system files that remain resident on the server”), each stall has its own set of necessary ingredients and equipment (the “set of associated system files required to execute the one or more applications”). The chef uses the local materials in the food stall rather than running back and forth to the central storage (“wherein some or all of the associated system files within a container stored in memory are utilized in place of the associated local system files that remain resident on the server”). The stall’s equipment and ingredients may be the same as what is in the central storage or specific to the stall’s cuisine (“said associated system files utilized in place of the associated local system files are copies or modified copies of the associated local system files that remain resident on the server”). To prevent cross-contamination, chefs do not share ingredients or equipment ("the application software cannot be shared between the plurality of secure containers of application software"), and each chef organizes their stall's ingredients and equipment as they choose ("each of the containers has a unique root file system that is different from an operating system's root file system").

While representative claim 1 recites computer-specific terminology such as "applications" and "containers" instead of chefs and stalls, "limiting the use of an abstract idea 'to a particular technological environment'" cannot render the claims patent eligible. *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 573 U.S. 208, 223 (2014) (citation omitted). The claims are directed to nothing more than the abstract idea of replicating and organizing data and applications into containers so applications can be used in different operating environments.

In performing this ***Step One*** analysis, courts often compare a claim to "the ideas found to be abstract in other cases before the Supreme Court and [the Federal Circuit]." *Intell. Ventures I LLC v. Cap. One Bank (USA)*, 792 F.3d 1363, 1367 (Fed. Cir. 2015). This is instructive here because courts have repeatedly held that schemes which are similar to the '814 Patent are abstract ideas. Here, the single method step recited in claim 1 is "storing in memory… a plurality of secure containers of application software." But merely storing data in a computer memory in a particular format is not patent eligible. *See, e.g.*, *Berkheimer v. HP Inc.*, 881 F.3d 1360, 1366 (Fed. Cir. 2018) (claims directed to "parsing, comparing, storing, and editing data" not patent-eligible); *Tenstreet, LLC v. DriverReach, LLC*, 826 F. App'x 925, 926 (Fed. Cir. 2020) (claims directed to "collecting, organizing, and storing data on a conventional computer network" not patent eligible); *Content Extraction*, 776 F.3d at 1345 (claims directed to "receiving" data, "recognizing" portions of that data, and "storing that information in a memory" not patent eligible).

It should be no surprise, then, that patents pertaining to organizing computer information into "containers" have routinely been invalidated as being directed to abstract ideas. For example, in *Hewlett Packard Co. v. ServiceNow, Inc.*, the patent-at-issue claimed "code for creating a hierarchy of derived containers, wherein a given derived container corresponds to," among other things, "a container definition node of an information model." No. 14-cv-00570-BLF, 2015 WL 1133244, at *2 (N.D. Cal. Mar. 10, 2015). The Court held that because "the container definition nodes and derived containers are nothing more than a data structure containing information for accessing the information repository hierarchically and a data structure for using that information," they were directed to the abstract idea of "categorizing and organizing information hierarchically." *Id.* at *7, *9. Here, too, claim 1 of the '814 Patent attempts to claim a data structure containing information, namely, a "container[] of application software" along with the system files on which the applications depend.

Similarly, in *Evolutionary Intel.m LLC v. Sprint Nextel Corp.*, the Federal Circuit invalidated claims from two patents directed to containers. 677 F. App'x 679, 680 (Fed. Cir. 2017) ("*Evolutionary II*"). The first patent recited a "computer-implemented method" including "searching… first container registers encapsulated and logically defined in a plurality of containers." *Evolutionary Intel., LLC v. Sprint Nextel Corp.*, 137 F. Supp. 3d 1157, 1161 (N.D. Cal. 2015) ("*Evolutionary I*"), *aff'd*, 677 F. App'x 679 (Fed. Cir. 2017). The second patent recited "[a]n apparatus… including a plurality of containers, each container being a logically defined data enclosure" comprising various components. *Id.* at 1161-62. The Federal Circuit affirmed the district court's determination that the claims were invalid for being "directed to the abstract idea of 'searching and processing containerized data.'" *Evolutionary II*, 677 F. App'x at 680.

The claims of the ’814 Patent are for “storing in memory… a plurality of secure containers” with various properties. As in *Evolutionary I*, those properties “are simply functional descriptions of conventional concepts of data processing… to govern the interaction of various data.” *Evolutionary I*, 137 F. Supp. 3d at 1168. That the containers are allegedly “secure” does not make the claim less abstract. To the extent the claim provides for “restricted access to resources,” that is still an abstract idea ineligible for patent protection. *Prism Techs. LLC v. T-Mobile USA, Inc.*, 696 F. App’x 1014, 1017 (Fed. Cir. 2017) (idea of receiving identity data and permitting access only if authorized is abstract).

3. The Asserted Claim Adds Nothing Inventive

Because claim 1 is directed to an abstract idea, it is unpatentable unless in *Alice* ***Step Two*** the Court determines that the claim adds a “sufficient” inventive concept that “transform[s] the abstract idea… into a patent-eligible application.” *Chamberlain Grp., Inc. v. Techtronic Indus. Co.*, 935 F.3d 1341, 1348-49 (Fed. Cir. 2019). It does not. Representative claim 1 merely recites generic, conventional computer components—for example, servers, operating systems, a processor, a kernel, local system files, applications, and objects—used to achieve the abstract idea of organizing data into containers so applications can be used in different operating environments.

The ’814 Patent does not contend that any of these components is individually novel, nor can it; these components were well known before the ’814 Patent was filed. ’814 patent, 1:20-3:19 (discussing these components in the background section). And each recited component performs its conventional function. Thus, the claimed computing functions and components, viewed individually and as an ordered combination, comprise the type of “basic functions of a computer” and “purely functional and generic” computer components that cannot supply an inventive concept. *Alice*, 573 U.S. at 225-26.

The structure of the claim also demonstrates the lack of an inventive concept. Despite a lengthy preamble, the claim is to "a method" with a single step: "storing in memory… a plurality of secure containers of application software." But the Federal Circuit has held that "storing… data" in memory "merely use[s] computers for their standard function[]." *Data Scape Ltd. v. W. Digital Corp.*, 816 F. App'x 461, 464 (Fed. Cir. 2020).

Further confirming the lack of inventive concept are cases holding that organizing data into "containers" does not add an inventive concept. In *Hewlett Packard*, the Court rejected claims directed to "container definition nodes" and "derived containers," explaining that "there is no inventive concept in combining computer readable media with the idea of categorizing and organizing information hierarchically." 2015 WL 1133244, at *9. And in *Evolutionary I*, the Court found the claims to "merely recite routine and conventional computer operations and structures as a means of implementing the abstract idea of searching and processing containerized data." 137 F. Supp. 3d at 1169.

At most, claim 1 describes the result of the abstract idea. In *Free Stream Media Corp. v. Alphonso Inc.*, the Federal Circuit held that such a result-oriented claim in a computing context is not "directed to an improvement of a technology." 996 F.3d 1355, 1365 (Fed. Cir. 2021). In *Free Stream*, the challenged patents claimed conventional computing devices that "constrain an executable environment in a security sandbox, and execute a sandboxed application in the executable environment." *Id.* at 1359. A "security sandbox," like the "secure containers" claimed here, "is a security mechanism for separating running programs." *Id.* The Federal Circuit rejected the patentee's argument that operating within this "sandbox" was a "specific improvement to the way computers operate." *Id.* at 1362 (citation omitted). Although the patentee had asserted that "its invention allows devices on the same network to communicate where such devices were previously unable to do so," the Court rejected that purported technical improvement because, like here, the claim "fail[ed] to recite a practical way of applying [the] underlying idea." *Id.* at 1363 (quoting *Interval Licensing LLC v. AOL, Inc.*, 896 F.3d 1335, 1343 (Fed. Cir. 2018)).

Like the claim in *Free Stream*, claim 1 here recites "storing in memory accessible to at least some of the server is a plurality of secure containers of application software," and then provides a laundry list of functional characteristics of the containers without reciting "'how' [each] functional result is achieved." *Free Stream,* 996 F.3d at 1363. Thus, although VirtaMove alleges that the containers are "secure" perhaps based on the limitation that "application software cannot be shared between the plurality of secure containers," this limitation cannot confer eligibility because it "do[es] not at all describe how that result is achieved." *Id.* at 1363-64.

4. The New, Conclusory Allegations In The Amended Complaint Do Not Make The Claims Patent Eligible

Rather that oppose Oracle's original motion to dismiss, VirtaMove amended its complaint to add three paragraphs related to the '814 Patent. First, VirtaMove makes the conclusory allegation that "[t]he technology in the '814 Patent was not routine or conventional." FAC, ¶ 12. This appears to be based on a 2009 Gartner report that the prior patent owner was one of several "Cool Vendors in Cloud Computing." Even accepting for purposes of this motion that AppZero was "cool," that does not make the '814 Patent's claims patentable.[1]

[1] To begin, "the § 101 inquiry must focus on the language of the Asserted Claims themselves," and VirtaMove does not plausibly allege that what made AppZero "cool" was that it practiced claim 1. *Mobile Acuity Ltd. v. Blippar Ltd.*, 110 F.4th 1280, 1293 (Fed. Cir. 2024); *Beteiro*, 104 F.4th at 1358. Even if it had, VirtaMove failure to allege that AppZero's purported "coolness" was due to something other than the abstract idea of organizing data into containers is fatal because "[t]he abstract idea itself cannot supply the inventive concept, no matter how groundbreaking the advance." *Brumfield, Tr. for Ascent Trust v. IBG LLC*, 97 F.4th 854, 868 (Fed. Cir. 2024); *ChargePoint*, 920 F.3d at 774.

Next, VirtaMove argues that because the Patent Office examined the '814 Patent application and determined that certain elements were not anticipated or "found collectively" in the prior art, the claims "cannot be described as 'routine' or 'conventional.'" FAC, ¶ 13. That is not the law. For literally every issued patent ever, an Examiner believed the proposed claims had novel elements or combinations and, thus, were not invalid under 35 U.S.C. §§ 102 or 103. That does not mean the claims also satisfy § 101. *Simio, LLC v. FlexSim Software Prods., Inc.*, 983 F.3d 1353, 1364 (Fed. Cir. 2020) ("the search for an inventive concept under § 101 is distinct from demonstrating novelty under § 102."). Were that the rule, no claim would ever be found directed to unpatentable subject matter because every claim was issued by an Examiner.

Finally, ignoring Oracle's food court analogy, the Amended Complaint asserts there is "no nontechnological analog" to the claims. FAC ¶ 14. As a matter of law, such "conclusory statements" or the allegation that "[t]he technology claimed is not directed to an abstract idea," are insufficient to overcome Oracle's motion. *Simio*, 983 F.3d at 1365. And even if the technological benefits VirtaMove alleges, *e.g.*, being "cost effective" and "more effectively shar[ing] a common compute platform" (FAC ¶ 14) were true, that does not make the claims patent eligible. This is because these alleged benefits merely "flow from performing an abstract idea" of replicating software components into containers. *BSG Tech LLC v. BuySeasons, Inc.*, 899 F.3d 1281, 1288 (Fed. Cir. 2018). Just as *BSG's* alleged improvements were "not improvements to database functionality" but instead "benefits that flow from performing an abstract idea in conjunction with a well-known database structure," VirtaMove's alleged improvements are merely benefits that flow from performing the abstract idea of replicating and organizing data into containers in a system of (well-known) servers. *Id.* "[P]lacing an abstract idea in the context of a computer" (replicating and organizing data into containers) "does not 'improve' the computer or convert the idea into a patent-eligible application of that idea." *Interval Licensing*, 896 F.3d at 1346.

Because VirtaMove's allegations merely "restate the claim elements and append a conclusory statement that" they are not abstract, routine or conventional, the amended complaint is insufficient to satisfy § 101. *Dropbox, Inc. v. Synchronoss Techs., Inc.*, 815 F. App'x 529, 538 (Fed. Cir. 2020).

5. The Remaining Claims Are Not Patent Eligible

Both the original and amended complaint specifically assert infringement for only claim 1 and do not allege that claims 2-34 provide a different or additional inventive concept. Thus, in its original motion, Oracle treated claim 1 as representative. *Health Discovery Corp. v. Intel Corp.*, 577 F. Supp. 3d 570, 577 (W.D. Tex. 2021) ("A district court may analyze representative claims for patent eligibility where all of the asserted and challenged claims are substantially similar and linked to the same purported abstract idea." (citing *Content Extraction*, 776 F.3d at 1348; *Cleveland Clinic Found. v. True Health Diagnostics LLC*, 859 F.3d 1352, 1360 (Fed. Cir. 2017))); *Berkheimer*, 881 F.3d at 1365 ("Courts may treat a claim as representative in certain situations, such as if the patentee does not present any meaningful argument for the distinctive significance of any claim limitations not found in the representative claim[.]").[2]

Despite knowing that for purposes of this motion Oracle considered claim 1 to be representative, the amended complaint does not allege that any other claim includes a different purported inventive concept. Because all claims are directed to the same abstract idea and none of the claims includes additional elements that transform that abstract idea into patentable subject

[2] Should VirtaMove attempt to introduce claims not specifically asserted in the Complaint in opposition to this motion, Oracle reserves the right to address those claims in its reply.

matter, all claims of the ’814 Patent are unpatentable for the same reason as representative claim 1.

B. The ’058 Patent Claims Are Invalid For Claiming Ineligible Subject Matter

Like the ’814 Patent, the ’058 Patent is invalid under Section 101. It fails ***Step One*** because the claims are directed to the abstract idea of replicating and organizing shared resources so they can be used by different entities for different purposes at the same time, and courts have routinely found similar claims unpatentable. It fails ***Step Two*** because it recites conventional computer components and does not add an inventive concept. The amended complaint does not cure this.

1. Background

The ’058 Patent has one independent claim, which is the only claim specifically asserted in the amended complaint. The claimed computing system includes a conventional prior art processor, operating system (“OS”), and shared library. The OS includes “critical system elements” (“CSEs”). The ’058 Patent admits that CSEs are prior art that include “[a]ny service or part of a service, ‘normally’ supplied by an operating system, that is critical to the operation of a software application.” ’058 Patent, 6:6-8 (defining CSEs); *id.*, 1:24-28 (noting use of CSEs in the prior art).

The ’058 Patent contends that because CSEs are ordinarily part of the operating system on which applications run (an “OSCSE”), there may be issues if two different applications attempt to use the same CSE at the same time or require different versions or configurations of the same CSE. ’058 Patent, 5:41-6:3. The ’058 Patent purports to solve this problem by “enabl[ing] the replication of critical system elements normally found in an operating system kernel. These replicated CSEs are then able to run in the context of a software application.” *Id.*, 5:21-24. In essence, rather than using only an OSCSE, the ’058 Patent “replicate[s]” the CSE for use by an application or set of applications. *Id.* The copy is called a shared library CSE (“SLCSE”). *Id.*, 2:27-30, 3:23-25.

Like the ’814 Patent, the ’058 Patent also is directed to an abstract idea. This is confirmed in the patent’s figures. “Prior Art” FIG. 2b “is illustrative of a known system architecture where critical system elements execute in user mode and execute in distinct context from applications in a single application process context” while “FIG. 3 is an architectural view of an embodiment of the invention.” ’058 Patent, 5:5-10.

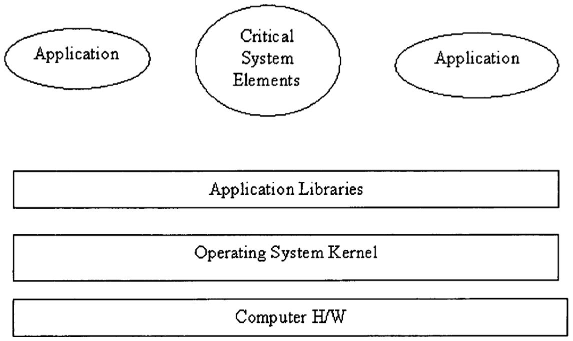


FIG. 2b
Prior Art

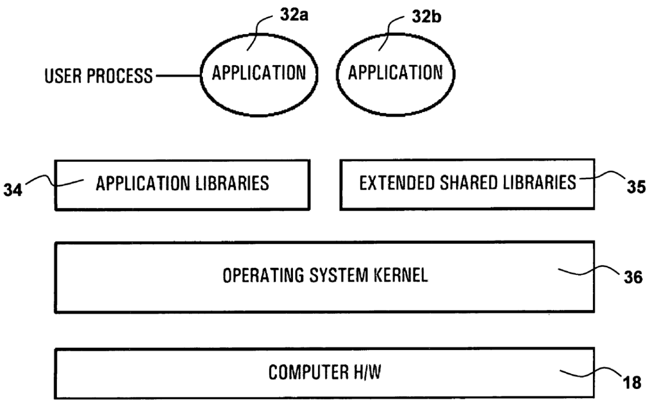
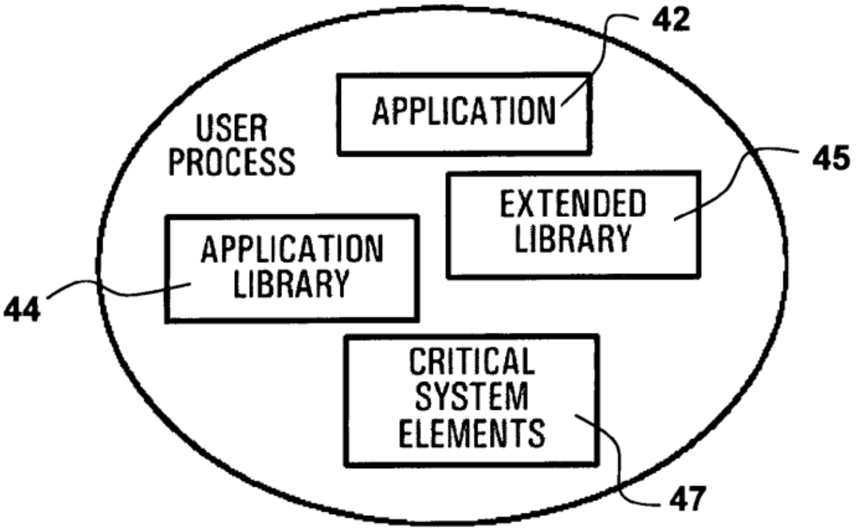


FIG. 3

Both the prior art and the alleged invention include the same components, but the alleged invention replicates and organizes the information. This is shown in the excerpt of FIG. 4, which illustrates the “User Process” (32a, 32b) of FIG. 3 to make clear that the invention merely copied certain prior art components (such as the critical system elements) into the user process:



2. The ’058 Patent Is Directed To An Abstract Idea

Representative claim 1, reproduced below, is directed to the abstract idea of replicating and organizing shared resources so they can be used by different entities for different purposes at the same time.

> 1. A computing system for executing a plurality of software applications comprising:
>
> a) a processor;
>
> b) an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor; and,
>
> c) a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode and
>
> i) wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications,
>
> ii) wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function, and
>
> iii) wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously.

’058 Patent, 10:51-11:14. The claimed “computing system” has only three elements—a conventional processor, operating system, and shared library. The prior art CSEs are replicated and organized to be stored in both the OS (as OSCSEs) and the shared library (as SLCSEs).

Although the claims of the ’058 Patent do not use the term “container,” their abstract idea can be illustrated using the same food court analogy described above. In the parlance of the ’058 Patent, there is a shared pantry (the “operating system”) with common ingredients (“OS critical system elements (OSCSEs)”). Each food stall, however, also has the specific equipment and ingredients needed to prepare its cuisine (“shared library having shared library critical system elements (SLCSEs)”). And, once again, various food stalls may have the same ingredients (*e.g.*, salt and pepper) as the shared pantry (“wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications”). But each chef may also use certain equipment or ingredients unique to their restaurant. For example, the Italian food stall would be the only one with a pizza oven and it is not shared with the other food stalls (“wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications”). In this way, each food stall can prepare food using similar but unique ingredients (“at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function”). By replicating common elements (stoves, pots, pans, *etc.*) in each food stall, the chefs can use that equipment (“SLCSEs”) to prepare food simultaneously (“wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously”).

The ’058 Patent is directed to the abstract idea of replicating and organizing shared resources so they can be used by different entities for different purposes at the same time, which, as shown in the cases cited in § III.A.2, courts have consistently found to be an abstract idea. If anything, the ’058 Patent is more abstract than the ’814 Patent because the ’058 Patent does not even purport to require “containers” or some other specific organizational structure.

The Federal Circuit’s decision in *Ericsson* makes clear that the “technical jargon” in the ’058 Patent does not save the patent from being directed to an abstract idea. *See Ericsson Inc. v. TCL Commc’n Tech. Holdings Ltd.*, 955 F.3d 1317, 1326 (Fed. Cir. 2020) (“Ericsson”). The *Ericsson* patent claimed “a method and system for limiting and controlling access to resources in a telecommunications system.” *Id.* at 1320. The Federal Circuit held that, although the claims were “written in technical jargon,” they were nevertheless “directed to the abstract idea of controlling access to, or limiting permission to, resources.” *Id.* at 1326. The Court rejected the patentee’s contention that the claims “solve the specific computer problem… of controlling app access in resource-constrained mobile phones” because the claims did not “ha[ve] the specificity required to transform a claim from one claiming only a result to one claiming a way of achieving it.” *Id.* at 1327-28 (citation omitted). Here, as in *Ericsson*, representative claim 1 is jargon rich, but merely recites conventional components (a processor, operating system, and a shared library), and three “wherein” clauses listing goals without claiming any specific “way of achieving” them. *Id.* at 1328.

Simply put, the problem that the ’058 Patent purports to address—the inefficiencies associated with multiple applications attempting to run using the same critical system elements—is age-old and not unique to the computer context. And its alleged solution—to replicate and organize system elements—is an unpatentable abstract idea.

3. The Asserted Claim Adds Nothing Inventive

The ’058 Patent fails *Alice* ***Step Two*** for reasons similar to the ’814 Patent. § III.A.3. Claim 1 recites generic computer elements—“processor,” “operating system[s],” “critical system elements,” “shared librar[ies],” and “applications”—used to achieve the abstract idea. ’058 Patent, 10:51-11:14. Neither the ’058 Patent nor VirtaMove’s Complaint alleges that any of these components was non-conventional, and the ’058 Patent acknowledges the components are prior art. For example, the components of a claim 1 are included in admitted “Prior Art” FIGs. 1, 2a, and 2b.

The structure of claim 1 illustrates that it lacks an inventive concept. As noted, the claim recites just three conventional computer components that are in the admitted prior art: (a) “a processor”; (b) “an operating system”; and (c) “a shared library”. *See* ’058 Patent, FIGs 1-2a (admitted prior art figures including each claimed component).

Although VirtaMove may argue the “ordered combination” of these conventional elements renders the claim eligible, each of the claim’s components performs the same function as it does in the prior art. *E.g.*, ’058 Patent, FIGs. 1-2b; 6:62-7:62; *see Free Stream Media*, 996 F.3d at 1365; *In re TLI Commc’ns LLC Pat. Litig.*, 823 F.3d 607, 615 (Fed. Cir. 2016) (where “recited physical components behave exactly as expected according to their ordinary use,” they do not constitute an inventive concept). Although the ’058 Patent purports to organize these conventional components in a novel manner, “[t]he abstract idea itself cannot supply the inventive concept, no matter how groundbreaking the advance.” *Brumfield*, 97 F.4th at 868; *ChargePoint*, 920 F.3d at 774.

4. The New, Conclusory Allegations In The Amended Complaint Do Not Make The Claims Patent Eligible

Once more, rather than oppose Oracle’s original motion to dismiss, VirtaMove amended its complaint to add three paragraphs related to the ’058 Patent. These paragraphs are essentially identical to those added concerning the ’814 Patent. *Compare*, FAC ¶¶ 12-14 and 23-25. As explained in § III.A.4, *supra*, the fact that AppZero is allegedly “cool,” that the Examiner allowed the asserted claims, and that VirtaMove has included conclusory assertions that the claims are not abstract and the components are not routine is not enough to meet § 101.

5. The Remaining Claims Are Not Patent Eligible

Claim 1 is the only specifically asserted claim in the amended complaint. In its original motion, Oracle explained that claim 1 was representative for purposes of this motion. VirtaMove amended the complaint but did not allege that claims 2-18 included patentable differences.

C. The Case Should Be Dismissed With Prejudice

VirtaMove reviewed Oracle’s original motion to dismiss and took its best shot to amend its complaint. The claims, however, remain unpatentable. Thus, Oracle seeks dismissal with prejudice because “it is clear that the defects [in VirtaMove’s amended complaint] are incurable.” *Great Plains Trust Co. v. Morgan Stanley Dean Witter & Co.*, 313 F.3d 305, 329 (5th Cir. 2002); *Data Scape,* 816 F. App’x at 464-65 (“We agree with the district court that [the plaintiff’s] ‘amendments do not remedy the deficiencies in its Complaint’ and that as such [the plaintiff] ‘failed to demonstrate that leave to amend would have been anything but futile.’”); *Sanderling Mgmt. Ltd. v. Snap Inc.*, 65 F.4th 698, 706 (Fed. Cir. 2023) (affirming dismissal under § 101 and agreeing that under circumstances of that case, “amendment of the complaint would have been futile”); *NexusCard, Inc. v. Kroger Co.*, 173 F. Supp. 3d 462, 468 (E.D. Tex. 2016), *aff'd*, 688 F. App'x 916 (Fed. Cir. 2017) (granting motion to dismiss under § 101 without leave to amend).

IV. <u>CONCLUSION</u>

Oracle respectfully requests that the Court dismiss VirtaMove’s amended complaint with prejudice.

Dated: April 25, 2025

Respectfully submitted,

/s/ Jared Bobrow

Jared Bobrow (admitted *pro hac vice*)
CA Bar No. 133712
jbobrow@orrick.com
Bas de Blank (admitted *pro hac vice*)
CA Bar No. 191487
basdeblank@orrick.com
Diana Rutowski (admitted *pro hac vice*)
CA Bar No. 233878
drutowski@orrick.com
Orrick, Herrington & Sutcliffe LLP
1000 Marsh Road
Menlo Park, CA 94025
Tel: (650) 614-7400
Fax: (650) 614-7401

Darryl J. Adams
Slayden Grubert Beard
401 Congress Avenue, Suite 1650
Austin, Texas 78701
Tel: 512-402-3550

Attorneys for Defendant Oracle Corp.

CERTIFICATE OF SERVICE

The undersigned certifies that on April 25, 2025, all counsel of record who are deemed to have consented to electronic service are being served with a copy of this document through the Court's CM/ECF system.

/s/ Jared Bobrow
Jared Bobrow